Fr. Conceicao Rodrigues College of Engineering,Mumbai

SOFTWARE ENGINEERING (CSC601)

Assignment -II

Date: 17-10-23

CO5: Identify risks, manage the change to assure quality in software projects.

Assignment 2

1. What is risk assessment in the context of software projects, and why is it essential?
2. Explain the concept of software configuration management and its role in ensuring project quality.
3. How do formal technical reviews (FTR) contribute to ensuring software quality and reliability?
4. Describe the process of conducting a formal walkthrough for a software project.
5. Why is it important to consider software reliability when analyzing potential risks in a project?

Rubrics :

Indicator	**Average**	**Good**	**Excellent**	**Marks**
Organization (2)	Readable with some mistakes and structured (1)	Readable with some mistakes and structured (1)	Very well written and structured (2)	
Level of content(4)	Minimal topics are covered with limited information (2)	Limited major topics with minor detailsare presented(3)	All major topics with minor details are covered (4)	
Depth and breadth of discussion(4)	Minimal points with missing information (1)	Relatively more points with information (2)	All points with in depth information(4)	
Total Marks(10)				

1. What is risk assessment in the context of software projects, and why is it essential?

Software Risk is actually a problem that may or may not occurs that shows the uncertainty of risks but if occur, unwanted losses or threatening or consequences will occur. It is generally caused due to a lack of incomplete information, control, or time. Risk Assessment and Risk Mitigation is a process in which identifying, assessing, and mitigating risk takes place to scope, schedule, cost, and quality of the project.

Risk assessment simply means to describe the overall process or method to identify risk and problem factors that might cause harm. It is actually a systematic examination of a task or project that you perform to simply identify significant risks, problems, hazards, and then to find out control measures that you will take to reduce risk. The best approach is to prepare a set of questions that can be answered by project managers in order to assess overall project risks.

Thus, the number of negative answers to these questions represents the severity of the impact of risk on the overall project. It is not about creating or making a large number of work papers, but rather simply identify and find out measures to control risks in your workplace.

2. Explain the concept of software configuration management and its role in ensuring project quality.

Whenever a software is build, there is always scope for improvement and those improvements brings changes in picture. Changes may be required to modify or update any existing solution or to create a new solution for a problem. Requirements keeps on changing on daily basis and so we need to keep on upgrading our systems based on the current requirements and needs to meet desired outputs. Changes should be analyzed before they are made to the existing system, recorded before they are implemented, reported to have details of before and after, and controlled in a manner that will improve quality and reduce error. This is where the need of System Configuration Management comes. System Configuration Management (SCM) is an arrangement of exercises which controls change by recognizing the items for change, setting up connections between those things, making/characterizing instruments for overseeing diverse variants, controlling the changes being executed in the current framework, inspecting and revealing/reporting on the changes made. It is essential to control the changes in light of the fact that if the changes are not checked legitimately then they may wind up undermining a well-run programming. In this way, SCM is a fundamental piece of all project management activities. Processes involved in SCM – Configuration management provides a disciplined environment for smooth control of work products. It involves the following activities:

Identification and Establishment – Identifying the configuration items from products that compose baselines at given points in time (a baseline is a set of mutually consistent Configuration Items, which has been formally reviewed and agreed upon, and serves as the basis of further development). Establishing relationship among items, creating a mechanism to manage multiple level of control and procedure for change management system.
Version control – Creating versions/specifications of the existing product to build new products from the help of SCM system. A description of version is given below:

3. How do formal technical reviews (FTR) contribute to ensuring software quality and reliability?

Formal Technical Review (FTR) is a software quality control activity performed by software engineers.

Objectives of formal technical review (FTR): Some of these are:

Useful to uncover error in logic, function and implementation for any representation of the software.
The purpose of FTR is to verify that the software meets specified requirements.
To ensure that software is represented according to predefined standards.
It helps to review the uniformity in software that is development in a uniform manner.
To makes the project more manageable.
In addition, the purpose of FTR is to enable junior engineer to observe the analysis, design, coding and testing approach more closely. FTR also works to promote back up and continuity become familiar with parts of software they might not have seen otherwise. Actually, FTR is a class of reviews that include walkthroughs, inspections, round robin reviews and other small group technical assessments of software. Each FTR is conducted as meeting and is considered successful only if it is properly planned, controlled and attended.

suppose during the development of the software without FTR design cost 10 units, coding cost 15 units and testing cost 10 units then the total cost till now is 35 units without maintenance but there was a quality issue because of bad design so to fix it we have to re design the software and final cost will become 70 units. that is why FTR is so helpful while developing the software.

The review meeting: Each review meeting should be held considering the following constraints-Involvement of people:

Between 3, 4 and 5 people should be involve in the review.
Advance preparation should occur but it should be very short that is at the most 2 hours of work for every person.
The short duration of the review meeting should be less than two hour. Gives these constraints, it should be clear that an FTR focuses on specific (and small) part of the overall software.
At the end of the review, all attendees of FTR must decide what to do.

Accept the product without any modification.
Reject the project due to serious error (Once corrected, another app need to be reviewed), or
Accept the product provisional (minor errors are encountered and should be corrected, but no additional review will be required).
The decision was made, with all FTR attendees completing a sign-of indicating their participation in the review and their agreement with the findings of the review team.

4. Describe the process of conducting a formal walkthrough for a software project.

The walkthrough is a review meeting process but it is different from the Inspection, as it does not involve any formal process i.e. it is a nonformal process. Basically, the walkthrough [review meeting process] is started by the Author of the code.

In the walkthrough, the code or document is read by the author, and others who are present in the meeting can note down the important points or can write notes on the defects and can give suggestions

about them. The walkthrough is an informal way of testing, no formal authority is been involved in this testing.

As there is an informal way of testing involved so there is no need for a moderator while performing a walkthrough. We can call a walkthrough an open-ended discussion, it does not focus on the documentation. Defect tracking is one of the challenging tasks in the walkthrough.

The technical review is a less formal way of reviewing a meeting process. It is process which is performed to give assurance about software quality. In Technical review process, the testing activity is performed by software engineers and other persons.

Here a work product is been inspected and reviewed for defects and other errors by individuals rather than the person who have produced it. The technical review is performed as a peer review without any management involved. So in context, the technical review varies from informal to quite formal.

Objectives of Technical Review:
Following are some of the objectives of Technical Review.

To create more reliable and manageable project.
To find technical error and defects.
To inform participants who have participated in technical review, about the technical content of document.
To maintain consistency of technical concepts.
To ensure that the software fulfils the requirements for which it is built.

5. Why is it important to consider software reliability when analyzing potential risks in a project?

The IEEE defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time." To most project and software development managers, reliability is equated to correctness, that is, they look to testing and the number of "bugs" found and fixed. While finding and fixing bugs discovered in testing is necessary to assure reliability, a better way is to develop a robust, high quality product through all of the stages of the software lifecycle. That is, the reliability of the delivered code is related to the quality of all of the processes and products of software development; the requirements documentation, the code, test plans, and testing.

Software reliability is not as well defined as hardware reliability, but the Software Assurance Technology Center (SATC) at NASA is striving to identify and apply metrics to software products that promote and assess reliability. This paper discusses how NASA projects, in conjunction with the SATC, are applying software metrics to improve the quality and reliability of software products. Reliability is a by-product of quality, and software quality can be measured. We will demonstrate how these quality metrics assist in the evaluation of software reliability. We conclude with a brief discussion of the metrics being applied by the SATC to evaluate the reliability .

Software cannot be seen nor touched, but it is essential to the successful use of computers. It is necessary that the reliability of software should be measured and evaluated, as it is in hardware. IEEE

610.12-1990 defines reliability as "The ability of a system or component to perform its required functions under stated conditions for a specified period of time." IEEE 982.1-1988 defines Software Reliability Management as "The process of optimizing the reliability of software through a program that emphasizes software error prevention, fault detection and removal, and the use of measurements to maximize reliability in light of project constraints such as resources, schedule and performance." Using these definitions, software reliability is comprised of three activities:

1. Error prevention
2. Fault detection and removal
3. Measurements to maximize reliability, specifically measures that support the first two activities

There has been extensive work in measuring reliability using mean time between failure and mean time to failure.[1] Successful modeling has been done to predict error rates and reliability.[1,2,3] These activities address the first and third aspects of reliability, identifying and removing faults so that the software works as expected with the specified reliability. These measurements have been successfully applied to software as well as hardware. But in this paper, we would like to take a different approach to software reliability, one that addresses the second aspect of reliability, error prevention.